



SEREEGA: Simulating event-related EEG activity

Laurens Krol, Juliane Pawlitzki, Fabien Lotte, Klaus Gramann, Thorsten Zander

► To cite this version:

Laurens Krol, Juliane Pawlitzki, Fabien Lotte, Klaus Gramann, Thorsten Zander. SEREEGA: Simulating event-related EEG activity. *Journal of Neuroscience Methods*, 2018, 309, pp.13-24. hal-01946076

HAL Id: hal-01946076

<https://inria.hal.science/hal-01946076>

Submitted on 5 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SEREEGA: Simulating Event-Related EEG Activity

Laurens R. Krol^{a,b,*}, Juliane Pawlitzki^{a,c}, Fabien Lotte^d, Klaus Gramann^{b,e,f}, and Thorsten O. Zander^{a,b,c}

^aTeam PhyPA, Biological Psychology and Neuroergonomics, Technische Universität Berlin, Berlin, Germany

^bBiological Psychology and Neuroergonomics, Technische Universität Berlin, Berlin, Germany

^cZander Laboratories B.V., Amsterdam, The Netherlands

^dInria, LaBRI (CNRS/University of Bordeaux/Bordeaux INP), Talence, France

^eCentre of Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

^fCenter for Advanced Neurological Engineering, University of California San Diego, USA

*Correspondence: lrkrol@gmail.com

Abstract

Background: Electroencephalography (EEG) is a popular method to monitor brain activity, but it is difficult to evaluate EEG-based analysis methods because no ground-truth brain activity is available for comparison. Therefore, in order to test and evaluate such methods, researchers often use simulated EEG data instead of actual EEG recordings. Simulated data can be used, among other things, to assess or compare signal processing and machine learning algorithms, to model EEG variabilities, and to design source reconstruction methods. **New method:** We present SEREEGA, *Simulating Event-Related EEG Activity*. SEREEGA is a free and open-source MATLAB-based toolbox dedicated to the generation of simulated epochs of EEG data. It is modular and extensible, at initial release supporting five different publicly available head models and capable of simulating multiple different types of signals mimicking brain activity. This paper presents the architecture and general workflow of this toolbox, as well as a simulated data set demonstrating some of its functions. The toolbox is available at <https://github.com/lrkrol/SEREEGA>. **Results:** The simulated data allows established analysis pipelines and classification methods to be applied and is capable of producing realistic results. **Comparison with existing methods:** Most simulated EEG is coded from scratch. The few open-source methods in existence focus on specific applications or signal types, such as connectivity. SEREEGA unifies the majority of past simulation methods reported in the literature into one toolbox. **Conclusion:** SEREEGA is a general-purpose toolbox to simulate ground-truth EEG data.

Keywords: Electroencephalography; Simulation; Evaluation; Event-Related; Classification; Brain-Computer Interface; Ground Truth

Highlights:

- Only simulated EEG data has a fully known ground truth, needed to validate methods.
- We present a general-purpose free and open-source toolbox to simulate EEG data.
- It is modular and currently includes five configurable head models and signal types.
- It supports noise, oscillations, event-related potentials, connectivity, and more.
- It provides a single framework to simulate EEG data with realistic analytic results.

1 Introduction

Having seen almost a century of continuous research and development since its first application on humans in the 1920s (Berger, 1929), electroencephalography (EEG) is now widely used in, among others, clinical settings, neuroscience, cognitive science, psychophysiology, and brain-computer interfacing, while its use continues to expand in fields such as neuroergonomics (Parasuraman & Rizzo, 2007; Frey, Daniel, Castet, Hachet, & Lotte, 2016), neurogaming (Krol, Freytag, & Zander, 2017), neuromarketing (Vecchiato et al., 2011), neuroadaptive technology (Zander, Krol, Birbaumer, & Gramann, 2016) and mobile brain/body imaging (Gramann et al., 2011). As of December 2017, PubMed reported over 140 000 publications related to EEG, with over 4 000 published in each of the past five years.

EEG reflects the electric fields that arise primarily due to the synchronous activity of post-synaptic potentials at apical dendrites in the cortical surface of the brain, as recorded by electrodes placed on the scalp. As such, it measures a specific subset of brain activity. This enables cognitive and affective correlates to be found in the EEG, allowing post-recording or even real-time evaluation of certain mental states exhibited by the recorded person, such as surprise (Donchin, 1981), error perception (Falkenstein, Hohnsbein, Hoormann, & Blanke, 1990; Blankertz, Schäfer, Dornhege, & Curio, 2002), task load (Klimesch, 1999; Mühl, Jeunet, & Lotte, 2014; Zander, Shetty, et al., 2017), or imagined movement (Pfurtscheller & Neuper, 2001; Blankertz, Dornhege, Krauledat, Müller, & Curio, 2007). Compared to other brain monitoring and imaging methods, EEG is relatively inexpensive and provides a high temporal resolution. It is also becoming increasingly portable and ready for personal use (Zander, Andreessen, et al., 2017) in various forms of neuroadaptive technology (Krol, Andreessen, & Zander, 2018). This explains EEG's relative popularity.

However, these advantages present a trade-off, with costs incurred primarily in spatial resolution. A single electrode records the average activity of up to a billion neurons, and probably never less than 10 million neurons (Nunez & Srinivasan, 2006). This and other issues including volume conduction, the placement and distance of the electrodes relative to the cortical generators of the activity they measure, and the complex relation between cortical functions and features of scalp potentials, require that great care is taken when analysing and interpreting raw EEG recordings.

A host of methods have been developed over the last decades to extract robust features from the recorded EEG that correlate to cortical functions and can be understood in a neurophysiological or statistical sense. Examples are the event-related potential technique (Luck, 2014), independent component analysis (Makeig, Jung, Bell, & Sejnowski, 1996), common spatial patterns (Guger, Ramoser, & Pfurtscheller, 2000), hierarchical linear modelling (Pernet, Chauveau, Gaspar, & Rousselet, 2011), and a many signal processing and machine learning algorithms (Lotte, Congedo, Lécuyer, Lamarche, & Arnaldi, 2007).

One major difficulty in developing techniques for EEG analyses is that there is no *ground truth* that describes the exact brain activity: The recorded EEG data cannot be compared to the actual neuro-electric activity of the brain because no technique exists to provide such a reference measurement. Thus, developers must use other ways to examine the validity of their EEG analysis approaches. The exact nature and detail of the required ground truth of course differs depending on the analysis.

To that end, simulated EEG data (or "toy data") has often been used to test and validate methods, as for example with blind source separation (Makeig, Jung, Ghahremani, & Sejnowski, 1999; Potter, Gadhok, & Kinsner, 2002), connectivity measures (Silfverhuth, Hintsala, Kortelainen, & Seppänen, 2012; Stam, Nolte, & Daffertshofer, 2007), artefact removal (Romo-Vazquez, Ranta, Louis-Dorr, & Maquin, 2007; He, Wilson, Russell, & Gerschütz, 2007), functional brain imaging (Gramfort, Strohmeier, Haueisen, Hamalainen, & Kowalski, 2011) and neurophysiological weight vector interpretation (Haufe et al., 2014). The authors of these examples all implemented simulation approaches from scratch, usually by linearly mixing a number of independent signals. This linear mixing was done using random weights, i.e., no realistic spatial information was taken into account. Other approaches use head models in order to provide more realistic linear mixing and add spatial dependencies to the simulation (e.g., Giraldo, den

Dekker, & Castellanos-Dominguez, 2010; Haufe, Tomioka, Nolte, Müller, & Kawanabe, 2010; Haufe, Nikulin, Müller, & Nolte, 2013; Huiskamp, 2008).

Such custom-made approaches are often difficult to reproduce, as they have been implemented using different software packages, are reported at different levels of abstraction, and/or may be using head models that are not publicly available.

Some authors have not implemented their own methods, but have instead relied on commercially available packages (e.g., Yao & Dewald, 2005; Lansbergen, van Dongen-Boomsma, Buitelaar, & Slaats-Willemse, 2011). These packages, however, are not specifically designed for the purpose of simulation, nor are they freely available to the general public or open to scrutiny.

One open-source package known to the authors that provides simulation functionality is the Source Information Flow Toolbox (SIFT; Delorme et al., 2011). Its main purpose is to investigate structural, functional, and effective connectivity between brain regions and networks, but the toolbox can also be used to simulate scalp EEG using a system of coupled oscillators. Such simulations can serve as effective ground truth for the same connectivity measures that the toolbox investigates, and have been used to evaluate blind source separation methods as well (Hsu, Mullen, Jung, & Cauwenberghs, 2014). The SIFT toolbox, however, is not intended for wider-purpose simulation and as such, the methods that can be meaningfully applied to the data it can generate are limited.

Another work allows a number of predefined responses (e.g. a P300, an eye blink, a frequency spike) to be configured along a given timeline of events (Lindgren, Merlini, Lecuyer, & Andriulli, 2018). It focuses on effects specific to brain-computer interface (BCI) applications and is aimed at the batch analysis of BCI classification algorithms.

In a more normative approach, Haufe and Ewald (2016) proposed a simulation and evaluation framework and made their code publicly available. The framework includes forward modelling using a realistic head model, both source and sensor noise with controlled signal-to-noise ratios, and signal generation based on autoregressive models. As with SIFT, this approach focuses solely on connectivity measures on continuous data, and thus provides no further signal generation options.

A final EEG simulation approach of note is the *phantom head* created by Oliveira, Schlink, Hairston, Knig, and Ferris (2016). They constructed a physical head model, filling a mannequin head with a conductive plaster and inserting antennae to simulate current dipoles. This allows unique effects to be investigated: For example, the authors investigated the effects of head motion by placing the model on a motion platform. Constructing such hardware, however, is not a viable approach for most questions where simulation can provide an answer. Importantly, software is easier to share, maintain, adapt, and extend.

Clearly, EEG data simulation is widely used as a tool to assess and validate the methods that are in use and in development. However, to the best of the authors' knowledge, there currently exists no software package whose sole or primary purpose it is to simulate different, configurable, and/or custom types of EEG data, i.e. a dedicated, general-purpose EEG data simulation tool. We therefore present SEREEGA, short for *Simulating Event-Related EEG Activity*, making EEG data simulation more accessible to researchers.

SEREEGA is a free and open-source MATLAB-based toolbox to generate simulated, event-related EEG data. Starting with a forward model obtained from a head model or pre-generated lead field, dipolar *brain components* can be defined. Each component has a specified position and orientation in the head model. Different activation patterns or *signals* can be assigned to these components. Scalp EEG data is simulated by projecting all signals from all components onto the scalp and summing these projections together.

SEREEGA is modular in that different head models and lead fields can be supported, as well as different activation signals. Five lead fields are currently supported, four pre-generated, the fifth customisable according to the user's needs from a standard head model. Five types of activation signals are provided, allowing the simulation of different types of systematic (event-related) activity in both the time and the frequency domain, as well as the inclusion of any already existing time series as an activation signal.

This toolbox is intended to be a tool to generate data with a known ground truth in order to evaluate neuroscientific and signal processing methods, such as blind source separation, source

localisation, connectivity measures, brain-computer interface classifier accuracy, derivative EEG measures, et cetera.

In the following, we first introduce the architecture of the toolbox and provide a brief introduction to its basic functionality and workflow. We then provide an analysis using established neuroscientific and brain-computer interfacing methods of a sample data set created with the toolbox.

2 SEREEGA Architecture and Functionality

2.1 Principles of EEG Simulation

To simulate EEG data, the toolbox solves the forward problem of EEG, prescribing how activation signals from specific sources in the brain are projected onto an array of electrodes on the scalp. In matrix notation, this can be written as

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \epsilon,$$

with \mathbf{x} denoting the vector of the recorded or simulated scalp signal, \mathbf{s} the source activation signal, \mathbf{A} the projection matrix used to project signals from the source to the scalp electrodes, and ϵ denoting a vector of noise.

In SEREEGA, the user defines \mathbf{s} , \mathbf{A} , and ϵ , allowing EEG data \mathbf{x} to be simulated.

Notation		Description
$\hat{\mathbf{A}} := \mathbf{A}_{\cdot, h, \cdot}$	$\in \mathbb{R}^{n_{chs} \times 3}$	Projection matrix of one source
\mathbf{A}	$\in \mathbb{R}^{n_{chs} \times n_{srcs} \times 3}$	Projection tensor of all sources
$e \in \{1, \dots, n_{eps}\}$	$\in \mathbb{N}$	One of n_{eps} epochs
\mathbf{E}	$\in \mathbb{R}^{n_{chs} \times n_{tps} \times n_{eps}}$	Simulated sensor noise
\mathbf{g}_c^k	$\in \mathbb{R}^{n_{tps}}$	Activation signal of one signal class c for one component k , $c \in \{1, \dots, n_{cls}^k\}$
$h \in \{1, \dots, n_{srcs}\}$	$\in \mathbb{N}$	One of n_{srcs} sources in the lead field
$k \in \{1, \dots, n_{cmps}\}$	$\in \mathbb{N}$	One of n_{cmps} brain components
n_{chs}	$\in \mathbb{N}$	Number of channels
n_{cls}^k	$\in \mathbb{N}$	Number of classes for component k
\mathbf{o}^h	$\in \mathbb{R}^3$	Orientation vector for source h
\mathbf{s}^k	$\in \mathbb{R}^{n_{tps}}$	Activation vector of component k
\mathbf{s}_t^k	$\in \mathbb{R}$	Activation of component k at time point t
$\hat{\mathbf{s}}_t^k$	$\in \mathbb{R}^3$	Oriented activation vector of \mathbf{s}_t^k
$t \in \{1, \dots, n_{tps}\}$	$\in \mathbb{N}$	One of n_{tps} time points per epoch
$\hat{\mathbf{x}}_t^k$	$\in \mathbb{R}^{n_{chs}}$	Projected activation of component k at time point t
$\hat{\mathbf{X}}^k$	$\in \mathbb{R}^{n_{chs} \times n_{tps}}$	Matrix of projected activation of component k
\mathbf{X}^e	$\in \mathbb{R}^{n_{chs} \times n_{tps}}$	Matrix of projected activations of one epoch e
\mathbf{X}	$\in \mathbb{R}^{n_{chs} \times n_{tps} \times n_{eps}}$	The final simulated EEG data at sensor level

Table 1: Notation used in Section 2.1.

Source activations are defined on the basis of so-called *components*: for each component, any number of different signal *classes* can be defined, which prescribe how corresponding activation signals are to be generated. For each signal class, a type of activation signal (for example, an event-related potential; see Section 2.3.3) is defined, as well as all corresponding parameters. Furthermore, for each component, a source from the lead field is specified, which is modelled as a dipole at a specific location in the brain. The component also contains this dipole's (i.e. this source's) orientation. A component thus prescribes what signal is to be simulated, and how it is to be projected onto the scalp.

SEREEGA simulates one segment of EEG data, or one *epoch*, at a time, and repeats this n_{eps} times to obtain a larger data set. In the following, we first consider a single epoch. The activation

signal of each class, $\mathbf{g}_c^k \in \mathbb{R}^{n_{tps}}$, is considered as a vector which consists of the activation signal (i.e. the amplitude time series) for all n_{tps} time points in this epoch.

Thus, for each component k , its activation \mathbf{s}^k consists of the summed activations of all signal classes \mathbf{g}_c^k assigned to that component, which are n_{cls}^k many:

$$\mathbf{s}^k = \sum_{c=1}^{n_{cls}^k} \mathbf{g}_c^k.$$

For most signal classes, its exact activation signal \mathbf{g}_c^k is determined procedurally at runtime based on the specified parameters.

We now describe how an activation signal is projected to the channels. For this, we consider the signal at a single time point t , by taking the t -th entry of \mathbf{s}^k , denoted by \mathbf{s}_t^k . We project this signal from a source denoted by h .

The activation signal is projected onto the electrodes on the scalp through a projection matrix \hat{A} which we obtain using lead field theory (Ferree & Clay, 2000). A lead field contains projection parameters that indicate for each electrode and source how an activation at that source is scaled when it is recorded at that electrode on the scalp. To that end, the activation signal is split in three directions (represented by the three base vectors of Euclidean space), resulting in a third-order tensor consisting of n_{chs} layers, n_{srcs} rows and 3 columns:

$$\mathbf{A} \in \mathbb{R}^{n_{chs} \times n_{srcs} \times 3}.$$

Each row describes the projection matrix for a single source. For a specific source h corresponding to a component k , $\hat{A} := \mathbf{A}_{\cdot, h, \cdot} \in \mathbb{R}^{n_{chs} \times 3}$ thus describes the projection matrix of that component. The orientation \mathbf{o}^h of source h can be expressed as a linear combination of the different base vectors contained in the three columns of \hat{A} :

$$\begin{aligned} (1, 0, 0)^T & \text{ the x-direction, pointing to the left ear,} \\ (0, 1, 0)^T & \text{ the y-direction, pointing to the nose, and} \\ (0, 0, 1)^T & \text{ the z-direction, pointing to the top of the head.} \end{aligned}$$

This can be expressed by

$$\mathbf{o}^h = o_1^h \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + o_2^h \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + o_3^h \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} o_1^h \\ o_2^h \\ o_3^h \end{pmatrix}, \quad \text{for } o_1^h, o_2^h, o_3^h \in \mathbb{R}.$$

To project a component's activation onto the scalp, this activation \mathbf{s}_t^k is first oriented by scaling it in the corresponding directions, yielding the oriented activation vector $\hat{\mathbf{s}}_t^k \in \mathbb{R}^3$:

$$\hat{\mathbf{s}}_t^k := \mathbf{s}_t^k \cdot \mathbf{o}^h = \begin{pmatrix} \mathbf{s}_t^k \cdot o_1^h \\ \mathbf{s}_t^k \cdot o_2^h \\ \mathbf{s}_t^k \cdot o_3^h \end{pmatrix}.$$

Then, the oriented activation vector $\hat{\mathbf{s}}_t^k$ is projected through the lead field which corresponds to the source of the activation, by multiplication with the projection matrix $\hat{A} = [a_{ij}]_{i=1, \dots, n_{chs}; j=1, \dots, 3}$:

$$\hat{\mathbf{x}}_t^k := \hat{A} \cdot \hat{\mathbf{s}}_t^k = \left[\sum_{j=1}^3 a_{ij} \hat{\mathbf{s}}_t^k \right]_{i=1, \dots, n_{chs}}.$$

This yields a vector $\hat{\mathbf{x}}_t^k \in \mathbb{R}^{n_{chs}}$ in which every element corresponds to the simulated signal amplitude at time point t , projected from source h to one electrode. To obtain the corresponding matrix for all time points for the component k , the vectors of all time points are concatenated:

$$\hat{X}^k = [\hat{\mathbf{x}}_1^k, \hat{\mathbf{x}}_2^k, \hat{\mathbf{x}}_3^k, \dots, \hat{\mathbf{x}}_{n_{tps}}^k].$$

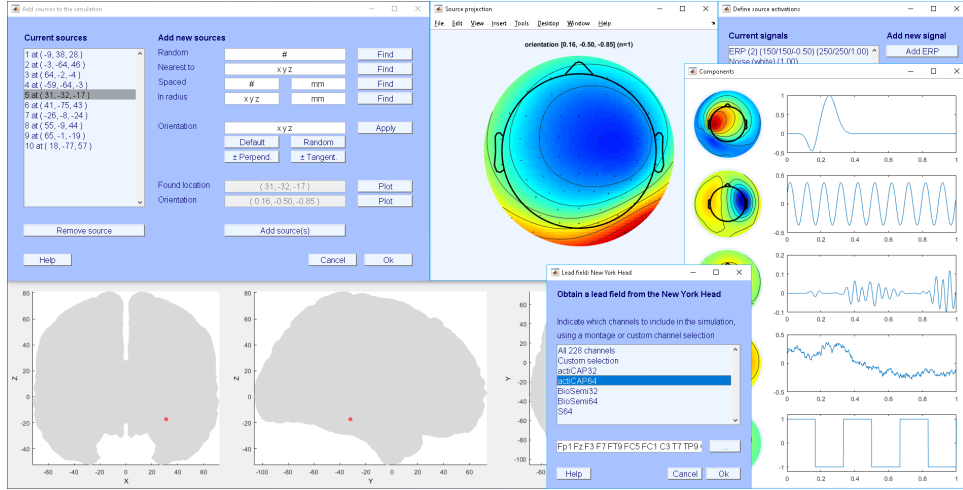


Figure 1: Impression of the SEREEGA GUI.

All simulated and projected activation signals \hat{X}^k of each component are summed to form one epoch. Thus, for one epoch, the simulated scalp signal X^e has the following form:

$$X^e = \sum_{k=1}^{n_{cmps}} \hat{X}^k.$$

The parameters in the signal classes describe how each component's activation signals vary between epochs. The projection matrix \hat{A} can also be made to vary from epoch to epoch, to simulate non-stationarities in the components' projections, for example due to shifts in electrode positions. Finally, all scalp signal matrices for all epochs are concatenated in the third dimension, yielding a third-order tensor in $\mathbb{R}^{n_{chs} \times n_{tps} \times n_{eps}}$. It is at this point that sensor noise \mathbf{E} is optionally added (source-level noise is defined as an element of \mathbf{g}_c^k). This yields the tensor of simulated EEG data:

$$\mathbf{X} = [X^1, X^2, X^3, \dots, X^{n_{eps}}] + \mathbf{E}.$$

This is the data format used by most software packages to represent epoched EEG data.

2.2 Platform and License

MATLAB R2014b or higher is recommended for SEREEGA. Some optional functions depend on the Digital Signal Processing (DSP) toolbox version 8.6 (R2014a) or higher. EEGLAB (Delorme & Makeig, 2004) is required as it is used for a number of functions. Lead field generation either requires additional head model files which can be downloaded from their respective websites, or the FieldTrip toolbox (Oostenveld, Fries, Maris, & Schoffelen, 2011). Since SEREEGA is modular, future functions may have further dependencies.

SEREEGA is licensed under the GNU General Public License, version 3, and the code is publicly available on GitHub¹. This manuscript refers to version 1.0.5-beta.

2.3 Terminology and Workflow

SEREEGA is available as an EEGLAB plug-in including a graphical user interface (GUI) that allows the core steps of designing and running a simulation to be performed; see figure 1. For more advanced use, SEREEGA is based on written commands and assignments.

¹<https://github.com/lrkrol/SEREEGA>

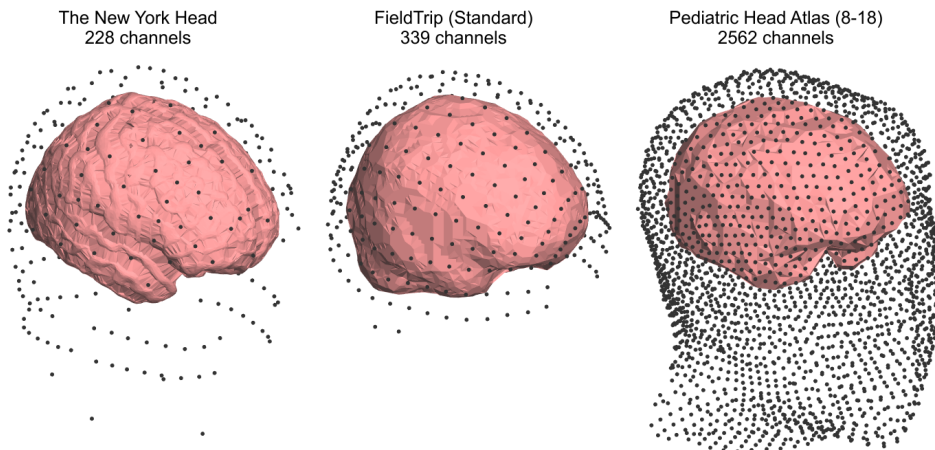


Figure 2: Channel locations and estimated brain boundaries for three head models. Left: The New York Head with its 228 channels. Middle: a 339-channel layout around a standard average head model available by default in FieldTrip. Right: The Pediatric Head Atlas (8 to 18, version 2) showing 2562 channels.

A general configuration variable holds information about the number of epochs to simulate, their length, and their sampling rate. This configuration, as well as many other variables, are contained within MATLAB structure arrays.

SEREEGA's forward model is contained in a *lead field* structure array. The lead field structure contains all possible sources within a virtual brain, modelled as dipoles at specific locations along with their projection patterns. The word *source* thus corresponds to an index in this lead field. For each such index, the lead field contains a position (x, y, z), and projection patterns along three axes (x, y, z). As mentioned in Section 2.1, a linear combination of the three projection patterns can effectively be used to virtually define the source dipole's *orientation* in 3D space. A source together with its orientation can be saved as a *component* structure array, which additionally needs to be assigned a *signal*: the simulated neuro-electrical activity that will be projected from it, such as an event-related potential or an oscillation at a specific frequency. These are defined separately and added to the components.

The main workflow consists of defining any number of such components, each containing at least one source, orientation, and signal. The simulation of scalp data finally generates each component's signals and projects them onto the scalp.

The following sections go through the main steps in more detail. Note that these steps must not necessarily be followed in this order.

2.3.1 Lead Field Generation

The lead field determines both the maximum possible number of sources as well as the number of channels that will be simulated. Currently, SEREEGA supports two processes to obtain a lead field: it can be obtained from an existing, pre-generated lead field, or a custom lead field can be generated from a given head model. Currently, support for four pre-generated lead fields is included.

The New York Head (ICBM-NY) pre-generated lead field includes almost 75 000 source locations and their projections onto 228 channels (Huang, Parra, & Haufe, 2016). The electrode positions follow the international 10-05 system (Oostenveld & Praamstra, 2001) but also include two rows of channels below the ears. Each source comes with a default orientation that orients it perpendicular to the cortical surface.

The Pediatric Head Atlases comprise three different head models with pre-generated lead fields for up to three different electrode layouts each, ranging from 128 to more than 2500 electrodes (Song et al., 2013). The models cover heads from three pediatric age clusters, 0-2, 4-8, and 8-18 years old. Lead field sources are spaced in an approximately $1 \times 1 \times 1$ mm grid and

range from 3188 to 4837 in number. These lead fields do not contain default dipole orientations. For inclusion in SEREEGA, the given electrode and dipole coordinates are transformed upon initialisation to be centred around (0, 0, 0) and aligned to the axes.

FieldTrip (Oostenveld et al., 2011) can be used to generate a lead field as needed. Using a given head volume, it can generate any number of sources with a given resolution, projecting to any number of channels. By default, a standard head model and a 339-channel definition file following the international 10–05 system are included. FieldTrip does not provide default source orientations.

When obtaining a lead field, it is possible to only use a subset of the available channels by indicating their labels. Convenience functions are available to simulate standard EEG cap layouts with e.g. 64 channels. Figure 2 shows the available channel locations for three head models.

2.3.2 Source Selection and Orientation

The lead field structure contains all possible sources. From these, one or more sources can be selected using different methods to be included in the simulation. It is possible to simply obtain a random source, or to obtain multiple random sources that are at least a given distance apart from each other. It is also possible to select the source nearest to a specific location in the brain, or all sources within a certain radius from a specific location or another source. Sources are referenced using their index in the lead field. Figure 3, left, shows the location of one source using EEGLAB's standard head model and corresponding plotting function. (In this case, EEGLAB's standard head model fits the lead field's head model. SEREEGA's default plotting functions do not depend on this being the case.)

Sources represent dipoles at the indicated location. Having selected a source location, this dipole can be oriented to face different directions, which determines how it projects onto the scalp. The lead field makes this possible by containing the dipole's projection pattern onto all selected electrodes into three mutually perpendicular directions x, y, and z—the base vectors mentioned in Section 2.1. A linear combination of these three projection patterns can then be used to effectively orient the dipole in space. Thus, a dipole's orientation is indicated using three values, with (1, 0, 0) representing a perfect x-positive orientation, (0, 1, 0) y-positive, $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ a 45-degree angle with the same amplitude, et cetera. Figure 3, right, shows the three separate patterns as well as a combined projection pattern for one source.

2.3.3 Signal Definition

Having selected a source location and orientation, i.e. a projection pattern, one or more *signals* can be defined to be projected onto the scalp as if they originate from that location. These are defined using *classes*. A class contains a signal's type and any parameters corresponding to that type in a structure array. For each simulated epoch, a signal will be procedurally generated using the parameters of that class.

SEREEGA currently includes four types of signals: systematic deflections in the time domain (i.e., event-related potentials), systematic modulations of oscillatory activity (i.e., event-related spectral perturbations), stochastic processes in the time domain (i.e., noise), and autoregressive signals (i.e., spatio-temporal signals with systematic interactions between them). Figure 4 provides an overview of three of these and their main parameters. A final *data* type enables the inclusion of any already existing time series as an activation signal. The following sections mention the primary free parameters for each type.

Event-Related Potentials An *event-related potential* (ERP; Luck, 2014) class defines one or more positive and/or negative 'peaks' or deviations from a baseline in sequence. Each peak is determined by its latency in ms, its width in ms, and its amplitude in μV . For example, the ERP in figure 4 is a single positive peak at 500 ms, 200 ms wide, of 1 μV .

A peak is generated by centring a normal probability density function around the indicated latency with the given width covering 6 standard deviations. This is then scaled to the indicated amplitude. For multiple peaks, each peak is generated individually and then summed together.

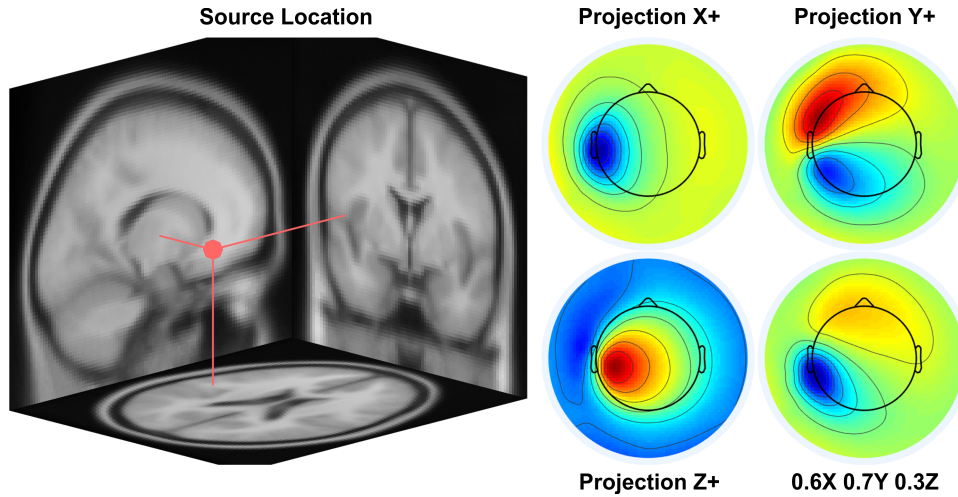


Figure 3: Left: The location of source number 7479 in the New York Head as plotted onto EEGLAB's standard head model. Right: Four projection patterns of that source onto the scalp. From top left to bottom right: projection in the x-positive, y-positive, and z-positive direction, and a linear combination of the three (arbitrary units). This effectively orients the dipole at the source's location in 3D space.

Event-Related Spectral Perturbations Where ERPs represent systematic activity in the time domain, *event-related spectral perturbation* (ERSP) classes are defined primarily using spectral features. The term ERSP refers to event-related spectral power changes (Makeig, Debener, Onton, & Delorme, 2004); in SEREEGA, it refers to the class of signals that generate specific spectral powers and/or changes therein.

First, a base frequency is defined, either as a pure sine wave of a single frequency, or as a frequency band. In this latter case, uniform white noise is band-pass filtered in the indicated band using a window-based finite impulse response filter with a Kaiser window and an automatically decided filter order (Oppenheim & Schaffer, 2010). The left middle panel of figure 4 shows an oscillation of 10 Hz, the panel to the right of that shows filtered noise with power focused between 5 and 10 Hz.

The minimal definition of an ERSP class consists of a frequency (or frequency band) and its amplitude. In case of a single frequency, a phase can optionally be indicated as well.

In other cases, these parameters merely define a base oscillation that is further modulated. The lower row of figure 4 presents two types of modulations that can be defined for ERSP classes: (inverse) burst modulation and amplitude modulation. The former attenuates or amplifies the signal to a given degree using a Tukey window of given width, latency, and tapering. A Tukey window can be adjusted ranging from a square to a cosine shape. This mimics event-related synchronisation and desynchronisation. The latter modulates the amplitude of the base signal to a given degree according to the phase of a sine wave, whose frequency and phase can be defined in the class. Additionally, a pre-stimulus period can be indicated such that the modulation only starts at a given latency.

Noise A class of a *noise* signal simply produces coloured noise, from either a normal Gaussian or a uniform distribution, of a given amplitude. A noise class is defined by at least a colour and an amplitude. The upper right panel of figure 4 shows white noise with an amplitude 1 μ V.

The Gaussian function uses MATLAB's DSP toolbox to generate noise with a spectral characteristic of $1/f^n$, with n either -2, -1, 0, 1, or 2. The uniform function first samples the signal from a uniform distribution and then applies a Fourier transformation to achieve the same spectral characteristic.

Data For the inclusion of any given pre-existing time series, for example data generated elsewhere or taken from existing EEG recordings, a *data* class can be defined. It must be given a matrix of data, epoch-dependent indices, and an absolute or relative amplitude. It will project the indicated data scaled to the given amplitude for each simulated epoch.

Autoregressive model A final type of class generates signals based on a linear *autoregressive model* (ARM), following the approach by Haufe and Ewald (2016). For a single source activation, this generates a time series where each sample depends linearly on its preceding samples plus a stochastic term. At least the order of the model, i.e. the number of non-zero weights on the preceding samples, and the signal amplitude must be configured. The exact weights of the model are determined automatically.

It is also possible to simulate connectivity between multiple sources. Uni- and bidirectional interactions can be indicated such that a time-delayed influence of one source on another is included in the signal. Either the number of interactions is configured and the toolbox will randomly select the interactions, or a manual configuration of the exact directionalities can be indicated. The model order is the same for all interactions.

Interacting signals are not simulated at runtime, but generated beforehand and included in the simulation as data classes.

2.3.4 Variability

It defeats the purpose of simulating multiple epochs if every epoch were to be exactly the same. To take an ERP as example, it may not be desirable to have a peak defined at 500 ms appear at exactly 500 ms in every epoch. In fact, perhaps the ERP should sometimes not appear at all. To that end, both classes and components contain parameters to add variability to the generated signal.

For all parameters displayed in figure 4 controlling the shape of the various signals, two more parameters exist that control those base parameters: each parameter has a corresponding *deviation* parameter, and a *slope* parameter.

The deviation indicates a range of allowed values. A base value of 2 with a deviation of 0.3, for example, will vary from epoch to epoch between 1.7 ($2 - 0.3$) and 2.3 ($2 + 0.3$) following a Gaussian function. Specifically, the range of deviation covers 6 standard deviations. This can be used to model general Gaussian variability within the system. A deviation is calculated for each parameter value separately for each epoch. For some parameters, one additional *shift* parameter is available to make these values deviate with equal magnitude within one epoch. This can be useful to e.g. have the latency of multiple peaks in an ERP vary together, maintaining the overall shape of the ERP.

The slope indicates a systematic change over time. A base value of 2 with a slope of -0.3 , and no deviation, will be 2 on the first epoch, and 1.7 ($2 - 0.3$) on the final epoch. In between, it will scale linearly with the number of the current epoch. Slopes can be used to model e.g. effects of fatigue and habituation, or to cover a range of intended values.

Each class also has a *probability* parameter that determines, for each epoch, that signal's probability of appearance. A probability of 0.5 indicates a 50% chance of the defined signal being generated for any simulated epoch. In case it is not generated, a flatline is returned instead, meaning it does not interfere with any other activations at that time.

Deviation and slope parameters are also available for components, to control the dipole orientation. Components can have one additional source of variability, with respect to the source location itself. Instead of one source index, multiple source indices can be indicated as that component's source. In that case, one of them will be randomly picked for each epoch. As such, together with the orientation variability, spatial variability can be added as well.

2.3.5 Simulating Scalp EEG

A *component* is a structure array that contains, for each element, at least one source index, a corresponding orientation, and at least one signal. Thus, by combining the results of the previous two steps into components, it is specified which of the defined signals are projected from which

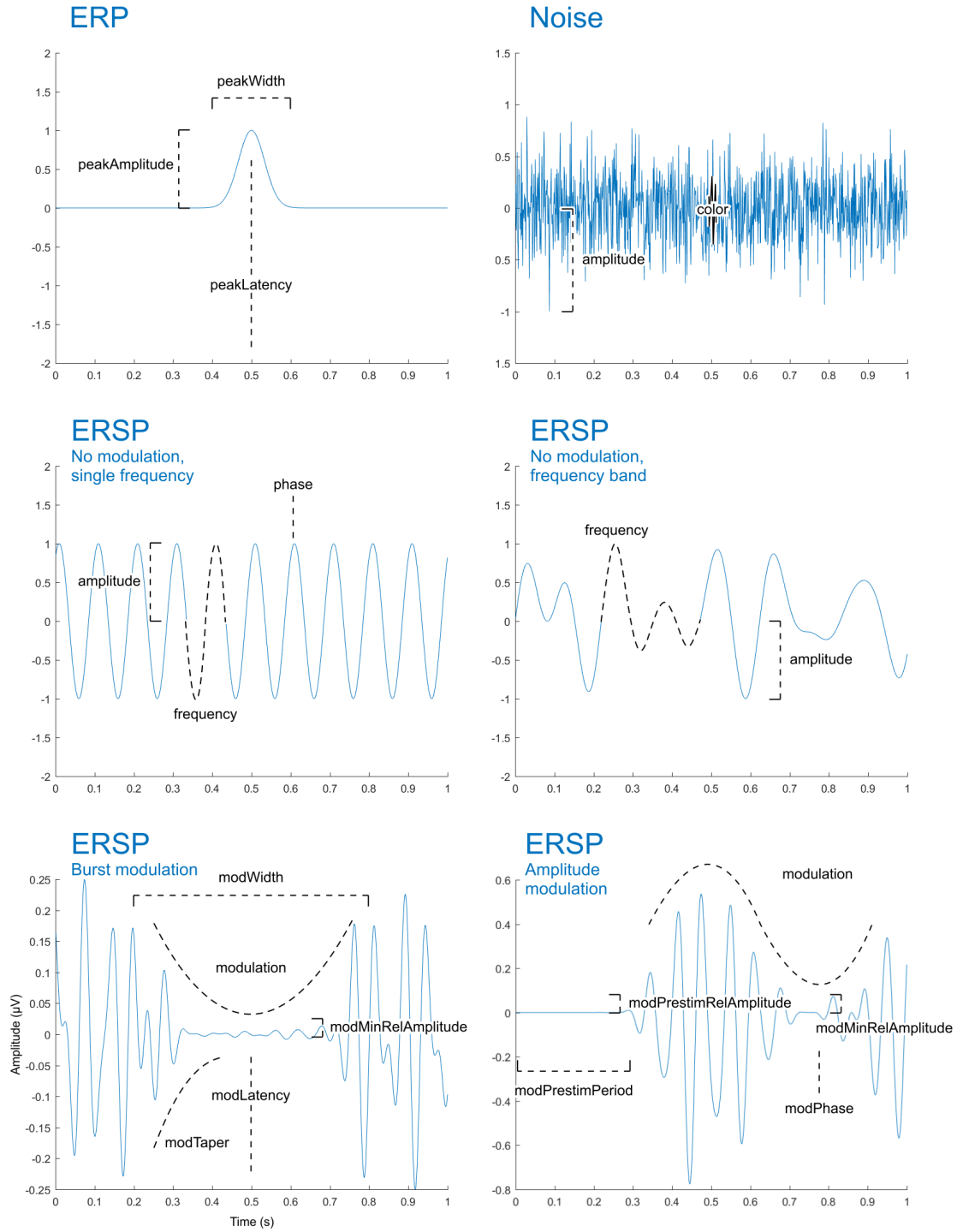


Figure 4: Sample signals annotated with the base variables that control their behaviour. From top left to bottom right: event-related potential (ERP); coloured noise; single-frequency unmodulated event-related spectral perturbation (ERSP); frequency band unmodulated ERSP; inverse burst-modulated ERSP; amplitude-modulated ERSP. Each base variable can additionally be given a deviation and a slope to control the epoch-to-epoch and systematic progressive variability of the signal. Not illustrated: the autoregressive model (ARM) and data classes.

source at which orientation. It is possible to assign more than one signal to a component. To add noise to an ERP, for example, both an ERP and a noise class can be added to the same component. All signals assigned to a component are summed together before being projected. When the same signal class is given to multiple components, a new instance of that signal will be generated for each component at runtime.

There is no limit to the number of components that can be defined, nor to the number of signal classes that can be added to each component.

A simulation function takes the defined components, the lead field, and the general configuration as input, and outputs the simulated scalp data in a *channels × samples × epochs* matrix, as well as *components × samples × epochs* source data. For each epoch, this function follows the forward model described in Section 2.1 to produce the scalp data. That is, it generates and sums each component's signals, projects them onto the scalp using the given lead field, source location, and orientation, and then sums all the projected activations. Each signal is generated independently of the others. For interacting signals, as with ARM classes, the signals are generated beforehand and included as data classes.

At this stage, a final layer of noise can optionally be added to the simulated scalp data to simulate sensor noise. This is uniform, temporally and spatially uncorrelated white noise.

2.3.6 Manual and Random Definition of Signal Classes

As a final point it is worth mentioning that not all of the available parameters must necessarily be set by the user. Only a small number are required, and a validation function automatically sets all others to their default values. All deviation-related values default to zero. Non-zero default values are chosen to be most useful; the phase of an oscillation, for example, defaults to a new random value for each epoch. A convenience function exists to add deviations and/or slopes of a certain percentage of each base value to signal classes. When the lead field contains default source orientations, these will be used automatically when no other orientation has been explicitly indicated.

When multiple components are to be simulated, it is also not necessary to manually define all signals. For one, the same signal can be added to multiple components. Also, convenience functions exist to generate any number of 'random' ERP or ERSP classes, based on a range of allowed base values.

2.3.7 Sample Code

The following lines of code reflect the workflow explained above, generating 100, 1-second epochs of brown noise from 64 sources spaced at least 25 mm apart, projected onto 64 channels.

```
epochs      = struct('n', 100, 'length', 1000, 'srate', 1000);
leadfield   = lf_generate_fromnyhead('montage', 'S64');
sourcelocs  = lf_get_source_spaced(leadfield, 64, 25);
signal      = struct('type', 'noise', 'color', 'brown', ...
                    'amplitude', 1, 'amplitudeDv', .5);
components  = utl_create_component(sourcelocs, signal, leadfield);
data        = generate_scalpdata(components, leadfield, epochs);
EEG         = utl_create_eeglabdataset(data, epochs, leadfield);
EEG         = utl_add_icaweights_toeeglabdataset(EEG, components, leadfield);
```

The `epochs` structure array contains three fields configuring the number of epochs, their length in ms, and the sampling rate in Hz. On the next line, a lead field is obtained from the New York Head using a preconfigured montage of 64 channels. Next, 64 source locations are randomly selected from the lead field, such that each source is at least 25 mm apart from all the others. As a signal, brown noise is used with an amplitude varying between 0.5 and 1.5 for each epoch. 64 components are then created by combining that same signal with the 64 previously-defined source locations. Because the New York Head is used, no orientations must be indicated explicitly: the toolbox will use the default values included in the lead field. Finally, scalp data is

generated: `generate_scalpdata` simulates the components' signals, projects them according to their respective orientations through the lead field, and generates the data set as configured in `epochs`. In the last two lines of code, the data set is transformed into an EEGLAB-compatible format, and a ground-truth ICA decomposition is added.

When using the GUI, these steps would be followed in more or less the same order.

3 Sample Data Set

To demonstrate some of the toolbox's basic functionality, we now present a sample data set that was generated using SEREEGA and simulates a number of known cortical effects during a hypothetical experiment. We then apply established analysis methods to this data and present the results.

3.1 Simulated Experiment and Cortical Effects

The hypothetical experiment represents a visual go/no-go experiment. The visual stimulus in question is a pattern change, eliciting a N70-P100-N135 complex over occipital sites (Pratt, 2011). In the 'go' condition, the pattern change represents a target stimulus eliciting a strong P3a-P3b complex, most pronounced over the parietal-central-frontal midline (Polich, 2007). Upon perception of the target stimulus, the participants initiate a manual response with their right hand, resulting in mu- and beta band desynchronisation over sites covering the contralateral motor cortex.

Specifically, the visually evoked potential complex was modelled after findings by Gomez Gonzalez, Clark, Fan, Luck, and Hillyard (1994) and Pratt (2011). The N70 was a single dipole projecting centrally from the visual cortex. The P100 and N135 projected bilaterally from Brodmann areas BA 17 and BA 18, respectively. The P3a-P3b was modelled after Polich (2007) and Debener, Makeig, Delorme, and Engel (2005) with dipole locations taken from the latter. The manual response, finally, mimics mu- and beta-band findings of manually responded visual targets by Makeig, Delorme, et al. (2004).

The visually evoked potentials were modelled as single-peak ERP classes centred around 70, 100, and 135 ms, respectively, with widths of 60, 60, and 100 ms, and amplitudes of 7.5, 7.5, and 10 μ V, as well as amplitude slopes of -2, -2, and -3 to simulate a level of habituation (Pratt, 2011).

The P3a is an ERP class with a single 400 ms wide peak at 300 ms of amplitude 10 μ V, and an amplitude slope of -5, simulating fatigue (Polich, 2007). The P3b ERP consists of two peaks. One at 400 ms (width 600 ms, amplitude 7 μ V, amplitude slope -5), and one at 500 ms (width 1000 ms, amplitude 2 μ V, amplitude slope -1). This latter peak represents the longer-lasting effects of the P3. The stronger slope of the P3b as compared to the P3a is a known finding reflecting habituation to the stimulus (Polich, 2007). All parameters of the ERP classes were given a deviation of 20% of their value relative to stimulus onset.

The motor cortex is simulated using four components for each class: both left and right motor cortex components are included, each with mu and beta band effects. In the non-target condition all components simply emit resting state mu (here as 8–12 Hz) and beta (19–26 Hz) activity implemented using the ERSP class with no modulation, and an amplitude of 2 μ V. In the target condition, the left motor cortex classes are given an inverse burst modulation centred around 650 ms (mu; width 600 ms, taper .5, relative amplitude .5) and 600 ms (beta; width 500 ms, taper .8, relative amplitude .5) respectively. ERSP parameters had a deviation of 10% relative to stimulus onset, with the exception of latency deviation, which was fixed at 100 ms.

To simulate background processes, brown noise was added to each component (as per Freeman, Ahlfors, & Menon, 2009), with an amplitude of 5 μ V. Furthermore, brown noise was also projected from a random selection of source locations, such that the total number of components in the data set was 64, all at least 25 mm apart from each other.

A single 'base' participant was defined and simulated using the specified component locations and orientations. For 10 subsequent participants, each component's location was randomly

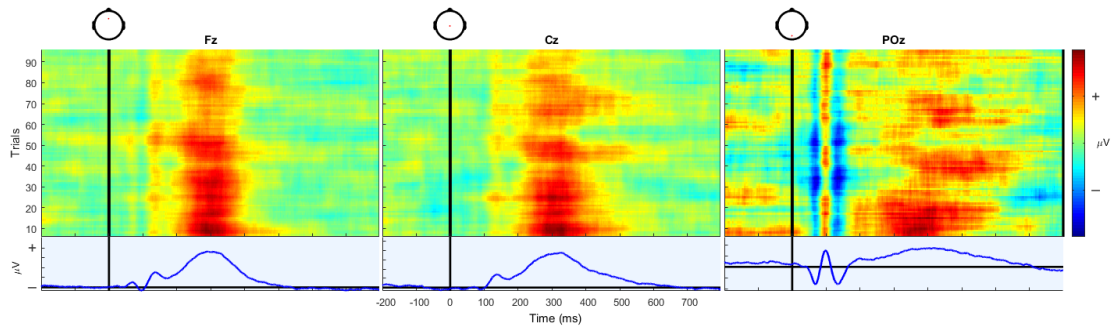


Figure 5: ERP image (Makeig, Debener, et al., 2004) at channel Fz, Cz, and POz showing the epoch-to-epoch variability of the simulated ERP signals as well as their spatial differences and effects over time.

chosen within 20 mm of the original locations. Dipole orientations were randomly varied to be within $\pm 25\%$ of the original value for each axis.

This data set was simulated using the New York Head Model and projected onto the S64 montage of 64 channels. For each condition (target versus non-target), 100 epochs were simulated of 2000 ms each with a 500 ms pre-stimulus period, at a sampling rate of 1000 Hz.

3.2 Analyses and Results

The generated data set was subjected to a number of standard analyses to demonstrate their applicability and to illustrate the data itself.

First, figure 5 shows a single-subject ERP image analysis at three midline electrodes of the ‘base participant’. This clearly shows the N70-P100-N135 complex, pronounced most strongly at parietal-occipital sites, as well as the P3, with the P3a most prominent over fronto-central sites and the P3b more pronounced further centrally and parietally. Also visible is the epoch-to-epoch variability of the ERP signals in time and in amplitude, as well as their spatial differences with peaks pronounced differently at different sites. The effects over time can also be seen, with amplitudes diminishing at different rates.

Second, a cluster-based group-level analysis was performed. Using DIPFIT (Oostenveld & Delorme, 2003), equivalent dipole locations were reconstructed from the data set’s ICA decomposition. These locations as well as the independent components’ calculated ERSPs were used for k-means clustering. Due to the variability of the data as well as the clustering algorithm, manual adjustment was still necessary to create one cluster containing all left motor cortex components. The calculated dipole locations are given in figure 6. As was defined, they vary around a given mean location. Figure 7 shows the mean calculated event-related spectral perturbation for this cluster for both the target (left) and the non-target (right) conditions. The target condition shows clear mu and beta effects similar to those found by Makeig, Delorme, et al. (2004, figure 8).

Finally, the data was subjected to an analysis of classifiability for a hypothetical BCI system to be able to detect the elevated P3 response in the target condition compared to the non-target condition, and, separately, the manual response.

The P3 classifier followed a windowed means approach (Blankertz, Lemm, Treder, Haufe, & Müller, 2011), using the mean amplitude of six consecutive 50 ms time windows starting 200 ms after stimulus onset as features. Before, the signal was bandpass-filtered between 0.1 and 15 Hz.

The manual response was classified using a logBP approach (Pfurtscheller & Neuper, 2001), using as features the power between 7 and 27 Hz, 400 to 1000 ms after stimulus onset, focused on sites covering the motor cortex: FC1, FC2, FC3, FC4, FC5, FC6, C1, C2, C3, C4, C5, and C6.

Both classifiers were implemented using BCILAB (Kothe & Makeig, 2013). A regularised linear discriminant analysis classifier was trained to separate classes (Duda, Hart, & Stork, 2001). A 5×5 nested cross-validation with margins of 5 was used to select the shrinkage regularisation

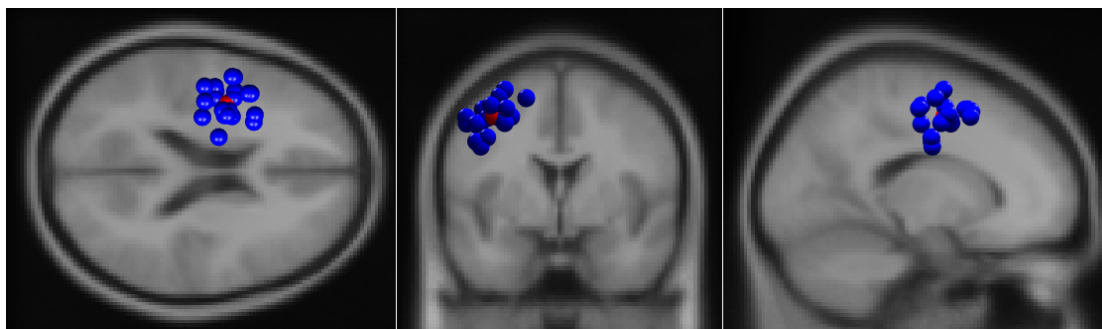


Figure 6: Locations of the left motor cortex equivalent dipoles in each simulated participant, from DIPFIT (Oostenveld & Delorme, 2003) calculations based on the independent components' scalp maps.

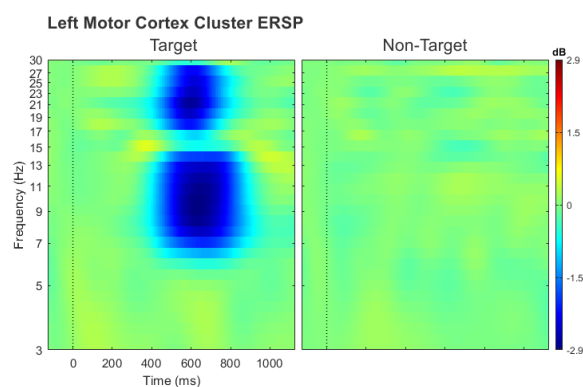


Figure 7: Mean event-related spectral perturbation calculated for the left motor cortex cluster. Left: target condition (right-manual response). Right: non-target condition (no response).

parameter, and to generate the estimates of the classifiers' accuracy listed in table 3.2. With chance level at 50% and significance reached at 57% (Müller-Putz, Scherer, Brunner, Leeb, & Pfurtscheller, 2008), we see that both classifiers reach significance for all 'participants', with the exception of participant 5, where no significant difference could be detected between the elicited P3 responses of the two conditions.

This shows the data's variability on another level and demonstrates that levels of significance can be achieved with neither a ceiling nor a floor effect.

Participant	WM TP	WM TN	WM Acc	BP TP	BP TN	BP Acc
0	0.78	0.75	0.76	0.70	0.64	0.67
1	0.76	0.69	0.72	0.67	0.69	0.68
2	0.74	0.74	0.74	0.57	0.58	0.57
3	0.77	0.72	0.74	0.61	0.62	0.61
4	0.72	0.57	0.64	0.65	0.64	0.64
5	0.55	0.58	0.56	0.66	0.69	0.67
6	0.68	0.57	0.62	0.61	0.70	0.65
7	0.71	0.64	0.67	0.70	0.68	0.69
8	0.71	0.61	0.66	0.63	0.55	0.59
9	0.76	0.67	0.71	0.56	0.59	0.57
10	0.67	0.57	0.62	0.69	0.64	0.66
Mean	0.72	0.65	0.69	0.64	0.64	0.64

Table 2: Classification accuracies for two classifiers, one using the windowed means approach (WM) and one based on band power (BP). TP = true positive rate, TN = true negative rate, Acc = overall accuracy. Significance is reached at an accuracy of 0.57. The WM classifier for participant 5 does not reach significance.

4 Discussion

In this paper we presented the core functionality of SEREEGA, a free and open-source MATLAB-based toolbox to simulate event-related EEG activity. SEREEGA allows any number of components to be defined in a virtual brain model, each with a specific location in the brain, a freely oriented scalp projection, and any number of signals. Five different types of activity are currently supported, allowing systematic effects in both the time and frequency domain to be simulated, mimicking known EEG features. Simulated EEG data can serve as ground truth for the development of new EEG analysis methods or the validation of existing ones.

The presented data set serves as a demonstration of some of SEREEGA's functions: a set of known cortical effects was simulated, mimicking a hypothetical experiment. It furthermore demonstrates how a basic initial configuration can be procedurally varied to simulate any number of additional participants with different anatomical layouts and variations in their cortical responses, leading to different results for example with respect to classification accuracies.

The classification accuracies presented here were only around 7 to 12 percentage points above significance on average. This is primarily due to the relative amplitude of the added noise, i.e., the signal-to-noise ratio of the generated data set. This can easily be varied to increase or decrease the classification accuracies as needed, or to investigate the accuracy as a function of this ratio. SEREEGA allows signal and noise components to be simulated separately, and then mixed to achieve a specific signal-to-noise ratio.

The ability to generate data as needed should be met with some caution. When validating signal detection mechanisms, for example, insufficiently considered parameters may lead to trivial results. Similarly, there is a potential danger of making circular arguments. For example, when SEREEGA is used to model a linear effect, and a linear method is subsequently used to again identify this effect, the obtained results could be essentially tautological. In such a case, the results would be applicable only to the extent that real EEG data contains a comparable linear effect. More generally, properties of signal simulation methods are based on assumptions that

must not necessarily, or not fully, be met in real EEG data, and this limits the transfer of findings from simulated data to real recordings. This is a fundamental property of all simulation approaches, and users of simulations should be aware of the generator algorithms underlying the data, how they relate to the methods they are investigating, and how they may or may not relate to real EEG recordings.

Indeed, the open architecture of any general-purpose toolbox places some responsibility with the user to make use of the tools in a way that is compatible with the tested hypothesis. Different analysis methods have different assumptions as to e.g. the number of active sources or the statistical properties of the source signals. While SEREEGA is capable of adhering to a large variety of assumptions, not all data that is produced by it will adhere to the same ones. This depends on the design of the data.

It is thus highly important to carefully consider a simulation's parameters. With SEREEGA, different levels of complexity and realism can be generated. For some purposes, there may be no explicit need to simulate complex data sets or otherwise precisely model specific known cortical effects: The results of the methods may hold regardless of the complexity of the simulation. For example, to validate, in theory, a method such as logBP, a pure sine wave among white noise in a low number of randomly mixed channels may suffice. However, other methods do require a higher level of realism with e.g. a higher number of channels and realistic spatial dependencies, a higher number of noise sources, et cetera. For the sake of demonstration, we opted for a more realistic hypothetical experiment to demonstrate in this paper. We do not mean to say however that this level of realism is the explicit goal of SEREEGA, nor that it is methodically preferable. We did intend to illustrate how SEREEGA *can* be used for this purpose.

It should also be mentioned that realism here refers to the ultimately simulated source and scalp signals, not their underlying neurophysiology. As such, for example, ERPs are simulated directly as ERP-like time series, not as e.g. lower-level phase-resetting oscillations. Similarly, other known EEG effects such as phase-amplitude coupling, travelling waves, EEG microstates, and certain cross-trial dependencies are also not currently implemented. Additional signal classes or component properties would have to be written before testing methods that require such types of signal generators. The online documentation provides a tutorial.

Another important factor in a simulation's level of realism is how realistic the noise is. The included stochastic noise processes follow current simulation standards, where in particular $1/f$ and $1/f^2$ frequency scales have been identified to characterise EEG (Bédard, Kröger, & Destexhe, 2006; Freeman et al., 2009). This is most likely a simplification. Empirical EEG recordings may have different spectral properties, especially non-stationarities over time, and are influenced by deeper and smaller processes whose activity is not directly measurable using EEG. However, there is currently no consensus regarding an alternative noise simulation method. In SEREEGA, any time series can be included in a simulation using the data class. This also allows noise extracted from empirical EEG recordings to be used. Where available, this may be a better solution. In the future, we may consider a public database of such recordings for all users to draw from. In fact, such a database could contain empirical recordings reflecting known cortical responses as well, allowing simulations to be pieced together by combining different parts of real EEG data. For this to be feasible, a large number of empirical data sets must be made available using a single, standardised format to describe them. Hierarchical Event Descriptor (HED) tags (Bigdely-Shamlo et al., 2016) offer a way to do that, and the Human Electrophysiology, Anatomic Data and Integrated Tools (HeadIT) Resource is one such database that could serve as precursor to the database needed for this purpose (Swartz Center for Computational Neuroscience (SCCN), n.d.).

Notable sources of artefacts are the eyes and muscles. The head models currently included in SEREEGA are cortical models and do not contain sources in the eyes or muscles. To the best of the authors' knowledge, no head model is currently available that does. A project is currently underway to construct a head model that contains a larger part of the human anatomy. For the moment, SEREEGA allows custom sources to be added to a simulation, by manually providing their coordinates and projection patterns. These can be taken from empirical recordings.

When taking data from empirical recordings, potential anatomical differences should be taken into account. Since SEREEGA supports different head models, it is not necessarily the case that

they use the same coordinate format (e.g. Talairach, MNI, or custom measurements). Sources found at the same $[x, y, z]$ location in different empirical studies or SEREEGA scripts may correspond to different actual locations depending on the head model used. Although authors who publish head models and lead fields often take great care to standardise their formats, they are not all uniform. Sometimes this is necessary: the dimensions of the Pediatric Head Atlas covering zero- to two-year-olds are of course smaller than those of an adult human's head model. SEREEGA includes plotting functions to investigate the size, shape, and location of a lead field's head model and the sources contained within, as well as to compare them against a standard adult head.

To the extent that their details were sufficiently reported, all (software) simulations that were performed by the authors mentioned in the introduction of this paper can be done using SEREEGA. The five signal types and other functions of SEREEGA cover and extend the processes used in those methods. This refers to the core steps of each simulation—further implementation details may of course differ. SIFT, for example, being focused on connectivity, has more options to control the parameters of the autoregressive models than SEREEGA currently does.

A limitation in the current architecture is that the defined components are necessarily independent at runtime: the procedurally generated activity of one component cannot presently depend on the activity in another. For such interactions to be modelled, the signals must be generated beforehand and included as data classes, as is the case for the autoregressive models.

In its current version, SEREEGA makes five accurate, high-density head models accessible in one toolbox, and allows five types of activation signals, enabling coloured noise, ERP-like deflections in the time domain, oscillatory activity with specific spectral characteristics and/or temporal modulations, and autoregressive signals to be simulated, as well as the inclusion of any other pre-generated or pre-recorded time series. This covers and extends the vast majority of past and present-day EEG simulation approaches. Additionally, FieldTrip-generated lead fields based on standard or custom head models can be used, and the toolbox's architecture allows it to be readily extended with additional head models and signals. For more specific needs, SEREEGA has been designed to be modular with extendability in mind.

SEREEGA thus represents, for the first time, an extensive general-purpose toolbox dedicated to simulating event-related EEG activity, making simulation methods more accessible, standardised, and reproducible.

With SEREEGA, we hope to help developers of EEG-based methods by making data simulation more accessible. With the consistent and increasing popularity of EEG, there is an accompanying need to further develop and validate EEG analysis methods. Simulated data can help with that by providing a ground truth to verify these methods' results. Seeing the current scope of application of simulated EEG studies, we believe that SEREEGA can help with the great majority of these, and can be easily extended for most of the remaining needs.

Acknowledgements

Part of this work was supported by the Deutsche Forschungsgemeinschaft (grant number ZA 821/3-1), Idex Bordeaux and LabEX CPU/SysNum, and the European Research Council with the BrainConquest project (grant number ERC-2016-STG-714567). The authors thank Mahta Mousavi for the band-pass filter design, Stefan Haufe for the autoregressive signal generation code and accompanying support, and Ramón Martínez-Cancino for his assistance with the GUI development.

Conflict of Interest

The authors declare no conflict of interest.

References

- Bédard, C., Kröger, H., & Destexhe, A. (2006). Does the $1/f$ frequency scaling of brain signals reflect self-organized critical states? *Physical Review Letters*, 97(11), 118102. doi: 10.1103/PhysRevLett.97.118102
- Berger, H. (1929). Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, 87, 527–570.
- Bigdely-Shamlo, N., Cockfield, J., Makeig, S., Rognon, T., La Valle, C., Miyakoshi, M., & Robbins, K. A. (2016). Hierarchical event descriptors (HED): Semi-structured tagging for real-world events in large-scale eeg. *Frontiers in Neuroinformatics*, 10, 42. doi: 10.3389/fninf.2016.00042
- Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.-R., & Curio, G. (2007). The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage*, 37(2), 539–550.
- Blankertz, B., Lemm, S., Treder, M. S., Haufe, S., & Müller, K.-R. (2011). Single-trial analysis and classification of ERP components – a tutorial. *NeuroImage*, 56(2), 814–825. doi: j.neuroimage.2010.06.048
- Blankertz, B., Schäfer, C., Dornhege, G., & Curio, G. (2002). Single Trial Detection of EEG Error Potentials: A Tool for Increasing BCI Transmission Rates. In J. R. Dorronsoro (Ed.), *Artificial Neural Networks – ICANN 2002* (Vol. 2415, pp. 1137–1143). Berlin, Germany: Springer.
- Debener, S., Makeig, S., Delorme, A., & Engel, A. K. (2005). What is novel in the novelty oddball paradigm? functional significance of the novelty P3 event-related potential as revealed by independent component analysis. *Cognitive Brain Research*, 22(3), 309–321. doi: 10.1016/j.cogbrainres.2004.09.006
- Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1), 9–21. doi: 10.1016/j.jneumeth.2003.10.009
- Delorme, A., Mullen, T., Kothe, C., Acar, Z. A., Bigdely-Shamlo, N., Vankov, A., & Makeig, S. (2011). EEGLAB, SIFT, NFT, BCILAB, and ERICA: New Tools for Advanced EEG Processing. *Computational Intelligence and Neuroscience*, 2011, 10:10–10:10. doi: 10.1155/2011/130714
- Donchin, E. (1981). Surprise!... Surprise? *Psychophysiology*, 18(5), 493–513.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). New York, NY: Wiley.
- Falkenstein, M., Hohnsbein, J., Hoormann, J., & Blanke, L. (1990). Effects of errors in choice reaction tasks on the ERP under focused and divided attention. In C. H. M. Brunia, A. W. K. Gaillard, & A. Kok (Eds.), *Psychophysiological Brain Research* (pp. 192–195). Tilburg, Netherlands: Tilburg University Press.
- Ferree, T., & Clay, M. (2000). *Lead field theory and the spatial sensitivity of scalp EEG* (Tech. Rep.). Eugene, OR, USA: Electrical Geodesics, Inc.
- Freeman, W. J., Ahlfors, S. P., & Menon, V. (2009). Combining fMRI with EEG and MEG in order to relate patterns of brain activity to cognition. *International Journal of Psychophysiology*, 73(1), 43–52. doi: 10.1016/j.ijpsycho.2008.12.019
- Frey, J., Daniel, M., Castet, J., Hachet, M., & Lotte, F. (2016). Framework for electroencephalography-based evaluation of user experience. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 2283–2294). New York, NY, USA: ACM. doi: 10.1145/2858036.2858525
- Giraldo, E., den Dekker, A. J., & Castellanos-Dominguez, G. (2010). Estimation of dynamic neural activity using a Kalman filter approach based on physiological models. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 2914–2917). doi: 10.1109/IEMBS.2010.5626281
- Gomez Gonzalez, C. M., Clark, V. P., Fan, S., Luck, S. J., & Hillyard, S. A. (1994). Sources of attention-sensitive visual event-related potentials. *Brain Topography*, 7(1), 41–51. doi: 10.1007/BF01184836

- Gramann, K., Gwin, J. T., Ferris, D. P., Oie, K., Jung, T.-P., Lin, C.-T., . . . Makeig, S. (2011). Cognition in action: imaging brain/body dynamics in mobile humans. *Reviews in the Neurosciences*, 22(6), 593–608.
- Gramfort, A., Strohmeier, D., Haueisen, J., Hamalainen, M., & Kowalski, M. (2011). Functional brain imaging with m/eeg using structured sparsity in time-frequency dictionaries. In G. Székely & H. K. Hahn (Eds.), *Information processing in medical imaging: 22nd international conference, ipmi 2011, kloster irsee, germany, july 3-8, 2011. proceedings* (pp. 600–611). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-22092-0_49
- Guger, C., Ramoser, H., & Pfurtscheller, G. (2000). Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, 8(4), 447–456. doi: 10.1109/86.895947
- Haufe, S., & Ewald, A. (2016). A simulation framework for benchmarking eeg-based brain connectivity estimation methodologies. *Brain Topography*. doi: 10.1007/s10548-016-0498-y
- Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., & Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87(0), 96–110. doi: 10.1016/j.neuroimage.2013.10.067
- Haufe, S., Nikulin, V. V., Müller, K.-R., & Nolte, G. (2013). A critical assessment of connectivity measures for EEG data: A simulation study. *NeuroImage*, 64(Supplement C), 120 - 133. doi: 10.1016/j.neuroimage.2012.09.036
- Haufe, S., Tomioka, R., Nolte, G., Müller, K. R., & Kawanabe, M. (2010). Modeling sparse connectivity between underlying brain sources for EEG/MEG. *IEEE Transactions on Biomedical Engineering*, 57(8), 1954–1963. doi: 10.1109/TBME.2010.2046325
- He, P., Wilson, G., Russell, C., & Gerschütz, M. (2007). Removal of ocular artifacts from the EEG: a comparison between time-domain regression method and adaptive filtering method using simulated data. *Medical & Biological Engineering & Computing*, 45(5), 495–503. doi: 10.1007/s11517-007-0179-9
- Hsu, S.-H., Mullen, T., Jung, T.-P., & Cauwenberghs, G. (2014). Online recursive independent component analysis for real-time source separation of high-density EEG. In *36th annual international conference of the IEEE engineering in medicine and biology society* (pp. 3845–3848). doi: 10.1109/EMBC.2014.6944462
- Huang, Y., Parra, L. C., & Haufe, S. (2016). The New York Head—a precise standardized volume conductor model for EEG source localization and tES targeting. *NeuroImage*, 140, 150–162. doi: 10.1016/j.neuroimage.2015.12.019
- Huiskamp, G. (2008). Interindividual variability of skull conductivity: an EEG-MEG analysis. *International Journal of Bioelectromagnetism*, 10(1), 25–30.
- Klimesch, W. (1999). EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2), 169–195.
- Kothe, C. A., & Makeig, S. (2013). BCILAB: a platform for brain-computer interface development. *Journal of Neural Engineering*, 10(5), 056014. doi: 10.1088/1741-2560/10/5/056014
- Krol, L. R., Andreessen, L. M., & Zander, T. O. (2018). Passive Brain-Computer Interfaces: A Perspective on Increased Interactivity. In C. S. Nam, A. Nijholt, & F. Lotte (Eds.), *Brain-Computer Interfaces Handbook: Technological and Theoretical Advances* (pp. 69–86). Boca Raton, FL, USA: CRC Press.
- Krol, L. R., Freytag, S.-C., & Zander, T. O. (2017). Meyendtris: A hands-free, multimodal Tetris clone using eye tracking and passive BCI for intuitive neuroadaptive gaming. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction* (pp. 433–437). New York, NY, USA: ACM. doi: 10.1145/3136755.3136805
- Lansbergen, M. M., van Dongen-Boomsma, M., Buitelaar, J. K., & Slaats-Willemse, D. (2011). ADHD and EEG-neurofeedback: a double-blind randomized placebo-controlled feasibility study. *Journal of Neural Transmission*, 118(2), 275–284. doi: 10.1007/s00702-010-0524-2
- Lindgren, J. T., Merlini, A., Lecuyer, A., & Andriulli, F. P. (2018). SimBCI - tool to simulate EEG and BCI. In *Proceedings of the 7th International Brain-Computer Interface Meeting*.

- Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4(2), 24.
- Luck, S. J. (2014). *An Introduction to the Event-Related Potential Technique* (Second ed.). Cambridge, MA, USA: MIT Press.
- Makeig, S., Debener, S., Onton, J., & Delorme, A. (2004). Mining event-related brain dynamics. *Trends in Cognitive Sciences*, 8(5), 204–210. doi: <http://dx.doi.org/10.1016/j.tics.2004.03.008>
- Makeig, S., Delorme, A., Westerfield, M., Jung, T.-P., Townsend, J., Courchesne, E., & Sejnowski, T. J. (2004). Electroencephalographic brain dynamics following manually responded visual targets. *PLOS Biology*, 2(6), 0747–0762. doi: 10.1371/journal.pbio.0020176
- Makeig, S., Jung, T.-P., Bell, A. J., & Sejnowski, T. J. (1996). Independent component analysis of electroencephalographic data. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems* (Vol. 8, p. 145–151). Cambridge, MA, USA: MIT Press.
- Makeig, S., Jung, T. P., Ghahremani, D., & Sejnowski, T. J. (1999). Independent component analysis of simulated ERP data. In T. Nakada (Ed.), *Integrated human brain science: Theory, method, applications (music)*. New York, NY: Elsevier.
- Mühl, C., Jeunet, C., & Lotte, F. (2014). EEG-based workload estimation across affective contexts. In A.-M. Brouwer, , T. O. Zander, & J. B. F. van Erp (Eds.), *Using neurophysiological signals that reflect cognitive or affective state* (pp. 227–241). Frontiers Media SA.
- Müller-Putz, G. R., Scherer, R., Brunner, C., Leeb, R., & Pfurtscheller, G. (2008). Better than random? A closer look on BCI results. *International Journal of Bioelectromagnetism*, 10(1), 52–55.
- Nunez, P. L., & Srinivasan, R. (2006). *Electric fields of the brain: the neurophysics of EEG*. New York, NY, USA: Oxford University Press.
- Oliveira, A. S., Schlink, B. R., Hairston, W. D., Knig, P., & Ferris, D. P. (2016). Induction and separation of motion artifacts in EEG data using a mobile phantom head device. *Journal of Neural Engineering*, 13(3), 036014. doi: 10.1088/1741-2560/13/3/036014
- Oostenveld, R., & Delorme, A. (2003). *DIPFIT 2.x*. Retrieved 2014-04-25, from <http://sccn.ucsd.edu/eeglab/plugins>
- Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J.-M. (2011). FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience*, 2011, 1:1–1:9. doi: 10.1155/2011/156869
- Oostenveld, R., & Praamstra, P. (2001). The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology*, 112(4), 713–719. doi: 10.1016/S1388-2457(00)00527-7
- Oppenheim, A. V., & Schaffer, R. W. (2010). *Discrete-time signal processing* (3rd ed.). London, UK: Pearson.
- Parasuraman, R., & Rizzo, M. (2007). *Neuroergonomics: the brain at work*. Oxford, UK: Oxford University Press.
- Pernet, C. R., Chauveau, N., Gaspar, C., & Rousselet, G. A. (2011). LIMO EEG: A toolbox for hierarchical linear modeling of electroencephalographic data. *Computational Intelligence and Neuroscience*, 2011, 3:1–3:11. doi: 10.1155/2011/831409
- Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7), 1123–1134. doi: 10.1109/5.939829
- Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. *Clinical Neurophysiology*, 118(10), 2128–2148. doi: 10.1016/j.clinph.2007.04.019
- Potter, M., Gadhok, N., & Kinsner, W. (2002). Separation performance of ICA on simulated EEG and ECG signals contaminated by noise. In *IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings (Cat. No.02CH37373)* (Vol. 2, pp. 1099–1104). doi: 10.1109/CCECE.2002.1013100

- Pratt, H. (2011). Sensory ERP components. In S. J. Luck & E. S. Kappenman (Eds.), *Oxford handbook of event-related potential components* (pp. 89–114). New York, NY: Oxford University Press. doi: 10.1093/oxfordhb/9780195374148.013.0050
- Romo-Vazquez, R., Ranta, R., Louis-Dorr, V., & Maquin, D. (2007). EEG ocular artefacts and noise removal. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5445–5448). doi: 10.1109/IEMBS.2007.4353577
- Silfverhuth, M. J., Hintsala, H., Kortelainen, J., & Seppänen, T. (2012). Experimental comparison of connectivity measures with simulated eeg signals. *Medical & Biological Engineering & Computing*, 50(7), 683–688. doi: 10.1007/s11517-012-0911-y
- Song, J., Morgan, K., Turovets, S., Li, K., Davey, C., Govyadinov, P., ... Tucker, D. M. (2013). Anatomically accurate head models and their derivatives for dense array EEG source localization. *Functional Neurology, Rehabilitation, and Ergonomics*, 3(2–3), 275–293.
- Stam, C. J., Nolte, G., & Daffertshofer, A. (2007). Phase lag index: Assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources. *Human Brain Mapping*, 28(11), 1178–1193. doi: 10.1002/hbm.20346
- Swartz Center for Computational Neuroscience (SCCN). (n.d.). *HeadIT: Human electrophysiology, anatomic data and integrated tools resource*. Retrieved 2018-07-25, from <http://headit.ucsd.edu>
- Vecchiato, G., Astolfi, L., De Vico Fallani, F., Toppi, J., Aloise, F., Bez, F., ... Babiloni, F. (2011). On the use of EEG or MEG brain imaging tools in neuromarketing research. *Computational Intelligence and Neuroscience*, 2011, 3:1–3:12. doi: 10.1155/2011/643489
- Yao, J., & Dewald, J. P. (2005). Evaluation of different cortical source localization methods using simulated and experimental EEG data. *NeuroImage*, 25(2), 369–382. doi: 10.1016/j.neuroimage.2004.11.036
- Zander, T. O., Andreessen, L. M., Berg, A., Bleuel, M., Pawlitzki, J., Zawallich, L., ... Gramann, K. (2017). Evaluation of a dry EEG system for application of passive brain-computer interfaces in autonomous driving. *Frontiers in Human Neuroscience*, 11, 78. doi: 10.3389/fnhum.2017.00078
- Zander, T. O., Krol, L. R., Birbaumer, N. P., & Gramann, K. (2016). Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity. *Proceedings of the National Academy of Sciences*, 113(52), 14898–14903. doi: 10.1073/pnas.1605155114
- Zander, T. O., Shetty, K., Lorenz, R., Leff, D. R., Krol, L. R., Darzi, A. W., ... Yang, G.-Z. (2017). Automated task load detection with electroencephalography: Towards passive brain-computer interfacing in robotic surgery. *Journal of Medical Robotics Research*, 2(1), 1750003. doi: 10.1142/S2424905X17500039